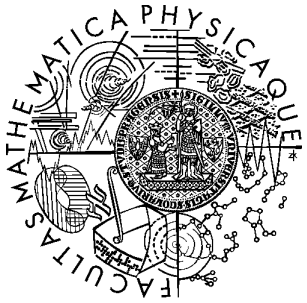
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁRSKÁ PRÁCA



Peter Beňa

Filmové titulky jako zdroj paralelních textů

Ústav formální a aplikované lingvistiky

Vedúci bakalárskej práce: Ing. Zdeněk Žabokrtský, Ph.D.
Studijní program: Informatika, Obecná informatika(IOI)

2007

Ďakujem vedúcemu bakalárskej práce Ing. Zdeňkovi Žabokrtskému, Ph.D. za odbornú pomoc pri vypracovaní bakalárskej práce, čas, ktorý mi venoval počas našich stretnutí a za priateľský prístup.

Prehlasujem, že som svoju bakalársku prácu napísal samostatne a výhradne s použitím citovaných prameňov. Súhlasím s požičiavaním práce a jej zverejňovaním.

V Prahe dňa Peter Beňa

Obsah

1 Spôsob riešenia práce **6**

2 Vylepšenia korpusu CzEng **7**
2.1 Make unzip generate 7
2.1.1 Make unzip 7
2.1.2 Make generate 7
2.2 Make solve_problem 8
2.2.1 Zmazanie súborov v nežiaducich jazykoch 9
2.2.2 Oprava názvov súborov 9
2.3 Make rough_clean 10
2.3.1 Zlievanie adresárov 11
2.3.2 Chyby v názvoch súborov 12
2.3.3 Súbory s podobnými názvami 13
2.3.4 Súbory, ktoré nemajú pár v druhom jazyku 14
2.3.5 Mazanie metainformácií a oprava I a l 15
2.3.6 Súbory s podobnými veľkosťami 17
2.3.7 Kontrola používania interpunkcie 20
2.3.8 Oprava interpunkcie 20
2.3.9 Mazanie súborov v iných jazykoch 21
2.3.10 Test na chýbajúcu diakritiku 22
2.4 Make toplain - prevod súborov do formátu plain 22
2.4.1 Nežiaduce znaky v texte 23
2.4.2 Strata údajov o čase zobrazenia 23
2.4.3 Počet súborov vo formáte plain 24
2.5 Make rename00 – premenovanie súborov k seriálom 25
2.6 Make del_dupl – zmazanie súborov k seriálom s rovnakými veľkosťami 25
2.7 Make select_pairs – výber najlepšieho páru 26

2.7.1 Odstránenie ID z názvov súborov 28
2.8 Make clean_after_pairing 28
2.8.1 Zmazanie párov s pomerom veľkostí mimo toleranciu 29
2.8.2 Prísnejší test na prítomnosť podobných slov 29
2.8.3 Oprava preklepov pomocou spellu a aspellu 30
2.9 Make finalclean – prečistenie súborov na konci 32

3 Možnosti na ďalšie rozšírenie projektu 33

4 Záver 35

Literatura 36

Názov práce: Filmové titulky jako zdroj paralelních textů
Autor: Peter Beňa
Katedra (ústav): Ústav formální a aplikované lingvistiky
Vedúci bakalárskej práce: Ing. Zdeněk Žabokrtský, Ph.D.
e-mail vedúceho: zabokrtsky@ufal.mff.cuni.cz

Abstrakt: Student se seznámí s metodami a nástroji pro budování paralelních korpusů a zaměří se na česko-anglický paralelní korpus Czeng. Hlavním cílem práce je zvýšit kvalitu té části Czengu, která byla vytvořena z anglických a českých titulků k filmům a seriálům. Především je nutné vypracovat automatické metody, které v paralelním korpusu naleznou a odstraní chybně spárované nebo jinak vadné texty nebo jejich části. Výsledky čištění korpusu budou kvantitativně vyhodnoceny.
Klíčová slova: Czeng, korpus, preklad, titulky

Title: Movie subtitles as a source of parallel texts
Author: Peter Beňa
Department: Institute of Formal and Applied Linguistics
Supervisor: Ing. Zdeněk Žabokrtský, Ph.D.
Supervisor's e-mail address: zabokrtsky@ufal.mff.cuni.cz

Abstract: After learning the basic principles of building parallel corpora, the student will focus on the Czech-English parallel corpus Czeng. The main goal of the work is to improve quality of the Czeng part created from Czech/English movie and series subtitles. Above all, it is necessary to design and implement methods for detecting wrongly aligned (or otherwise problematic) subtitle files or their parts. Impact of the cleaning methods on the corpus quality will be evaluated quantitatively.
Keywords: Czeng, corpus, translation, subtitles

Kapitola 1

Spôsob riešenia práce

Ako zo zadania práce vyplýva, hlavným cieľom je odstrániť z paralelného korpusu Czeng chybne spárované súbory, prípadne súbory, ktoré obsahujú chyby, pre ktoré ich nie je možné v korpuse ponechať. Aby sme tieto ciele splnili, budeme musieť do programu, ktorý automaticky spracováva data s titulkami k filmom a seriálom, pridať ďalšie testy alebo pôvodné testy upraviť.

V prípade nových testov si ale musíme uvedomiť, kam v programe máme nový test zaradiť. Možno by sa dalo predpokladať, že väčšina nových testov bude zaradená až za výber najlepšieho páru, aby sme tak odstránili chybné páry. Problém však budeme riešiť trochu odlišne. Ak by sme totiž išli cestou odstraňovania chybných párov, zaručene by sme strácali najlepšie páry vo výsledných datach. Pritom by mohlo byť efektívnejšie v chybnom páre odstrániť problematický súbor a nahradiť ho iným, aby sme tak zachovali pár. Odstraňovať chybné súbory však budeme ešte pred výberom najlepšieho páru, aby sme nemuseli hľadať najlepšie páry zakaždým, keď objavíme chybný pár.

To znamená, že testy, ktoré bude možné vzhľadom na ich časovú zložitosť spustiť na väčších datach, spustíme ešte pred výberom najlepšieho páru. Tým zabránime tomu, aby sa problematické súbory dostali do najlepších párov a tieto neskôr celé zahodili. Taktiež týmto spôsobom znížime časovú zložitosť výberu najlepších párov.

Kapitola 2

Vylepšenia korpusu CzEng

2.1 Make unzip generate

2.1.1 Make unzip

Pri rozpakovaní archívov so súbormi z portálov titulky.com a opensubtitles.org sa používajú jednoduché príkazy, pri ktorých je malý priestor na možné vylepšenia. Pri vytváraní archívov sa zvažovalo, akým spôsobom sa data skomprimujú. Rozhodovalo sa medzi rýchlosťou kompresie a dekompresie a úsporou miesta na disku. Nakoniec sa zvolil rýchly kompresný algoritmus aj napriek tomu, že skomprimované data sú asi o 30% väčšie ako by boli pri skomprimovaní do formátu bzip2. Nakoľko sa ale dá projekt spustiť aj po skopírovaní zdrojových kódov na harddisk bez nutnosti kopírovania skomprimovaných dát, nie je veľkosť archívov pri použití DVD nosiča problematická. Rozpakované data zaberajú v oboch prípadoch rovnako veľký priestor na disku. Úspora času je teda v tomto prípade rozhodujúca.

Táto časť projektu sa teda meniť nebude. Optimalizácia veľkosti spakovaných dát nie je nutná pri ponechaní archívov na DVD. Na rýchlosť bola táto časť projektu optimalizovaná už pri jej vývoji.

2.1.2 Make generate

0.5cmKeď sa v stromovej štruktúre vytvárajú linky na súbory s datami, môže dôjsť k chybám. Častokrát nie je dostupné číslo CD alebo rok natočenia filmu, prípadne seriálu. Neskoršie testy prípadné chyby nájdu a zle

pomenované linky odstránia za predpokladu, že ich názvy nezodpovedajú jednému z dvoch regulárnych výrazov (jeden pre filmy a druhý pre seriály). Ak sú niektoré údaje v názvoch v nesprávnom poradí, môže dôjsť k chybe (názov linku nezodpovedá regulárnemu výrazu). Chýbajúci rok natočenia filmu sa dá v niektorých prípadoch zistiť z tabuľky, ktorú využívame pri vytváraní mien súborov z portálu opensubtitles.org. V prípade nesprávneho poradia údajov v názve linku môžeme jednotlivé údaje od seba oddeliť a otestovať pomocou regulárnych výrazov, o aký údaj ide. Následne vieme údaje zoradiť v správnom poradí.

Otázkou samozrejme ostáva kedy je najefektívnejšie opravovať chyby, ktoré vzniknú pri vytváraní názvov linkov. Mohli by sme sa chyby pokúsiť odhaľovať chyby už pri vytváraní mien. To by ale znamenalo, že by sme pri každom súbore testovali, či niektorý údaj nechýba a následne kontrolovali, či opravený názov zodpovedá regulárnemu výrazu. Ak budeme testovať len súbory, ktorých názvy neodpovedajú žiadnemu platnému regulárnemu výrazu, nemusíme hľadať chyby v názvoch, ktoré sú v poriadku. Počet testovaných súborov sa tak zníži asi na 10% celkového počtu.

Ak by sme sa pokúšali chyby hľadať ihneď, mohol by nastať problém v prípade, že k danému súboru nepoznáme jazyk. Takéto súbory sa rýchlo presunú do adresára *problem*. V adresári *problem* dokážeme zmazať súbory so zadaným jazykom iným ako je jazyk český alebo anglický. Zvyšné súbory v adresári *problem* sú prevažne v českom jazyku. Ak budeme opravovať chyby v názvoch súborov až v adresári *problem*, získame zoznam súborov bez zadaného jazyka, ktorých obsah by mal byť v českom jazyku. Pri skoršom testovaní by sme túto chybu nemuseli odhadom opraviť. Stále totiž pripúšťame, že niekoľko súborov, ktoré označíme za súbory s českým titulkami, budú obsahovať text v inom jazyku. Ak sa ale dopustíme takejto chyby, neskoršie testy by ju mali odstrániť.

2.2 Make solve_problem

Po rozpakovaní archívov s datami sa súbory premenovali tak, aby sa v ich názvoch nachádzali všetky požadované údaje. Tvary názvov súborov pre filmy a seriály boli definované pomocou regulárnych výrazov. Počas premenovávania súborov však mohli byť detekované chyby a link na súbor bol vytvorený v adresári *problem* (na jednej úrovni s adresármi *movies* a *series*).

Súbory v adresári *problem* väčšinou nemajú zadaný jazyk, chýba číslo CD alebo rok natočenia filmu. Kód jazyka uvedený v názve súboru prípadne nie je kódom anglického ani českého jazyka (kódy *en* a *cs*).

2.2.1 Zmazanie súborov v nežiaducich jazykoch

Ako prvé sú z adresára zmazané všetky súbory, ktoré majú v názve dvoj- až trojznakový kód. Pri kontrole súborov v náhodne vybratej vzorke sa zistilo, že súbory s iným ako požadovaným kódom obsahujú text v jazyku, aký je uvádzaný. Pritom ide prevažne o súbory v slovenskom jazyku. Testovanie na výskyt textu v českom, prípadne anglickom jazyku by neprinieslo požadovaný efekt, preto sú všetky súbory so zadaným jazykom zmazané. Linky na súbory s českým a anglickým kódom sa v adresári *problem* určite nenachádzajú. Pri premenovávaní súborov sme vytvárali linky na české a anglické súbory iba v adresároch *movies* a *series*. Preto zmazanie všetkých súborov s potenciálnym kódom jazyka nezmaže súbory, ktoré by v názve obsahovali kód českého alebo anglického jazyka.

Takýmto spôsobom sa nám podarilo zmazať 3 040 súborov. Tieto by sa do najlepších párov aj tak nedostali, nakoľko sa pred tým adresár *problem* nespracovával a všetky súbory sa tak pri záverečnom mazaní zahodili. Tým, že sme ale tieto súbory odstránili, umožnili sme tak ďalšie spracovanie súborov v adresári *problem*.

2.2.2 Oprava názvov súborov

Po predošlom mazaní ostali v adresári *problem* súbory bez kódu jazyka v názve. Náhodným výberom skontrolujeme, v akom jazyku je obsah týchto súborov napísaný. Nakoľko boli všetky vybrané súbory v českom jazyku, označíme všetky súbory za české a budeme očakávať, že testy na prítomnosť diakritiky a najpoužívanejších českých hlások prípadné chyby odhalia.

Pri vytváraní linkov na súbory z portálu www.opensubtitles.org sme mali k dispozícii tabuľky, v ktorých sa na jednom riadku nachádzalo ID filmu, názov filmu a rok natočenia filmu. V adresári *problem* sa nachádzajú aj linky na súbory z portálu www.titulky.com. Mená súborov pritom vytvárali samotní autori a niektoré údaje ako číslo CD a rok natočenia filmu v názve súboru môžu chýbať. Rok natočenia filmu ale v niektorých prípadoch môžeme

uhádnuť. Z tabuľky si načítame dvojice názov filmu a rok natočenia filmu. Názov filmu upravíme, konkrétne z neho vymažeme spojky, predložky, číslice a medzery. Do asociatívneho poľa (hashu) potom uložíme pre daný film na základe praveného názvu údaj o roku natočenia daného filmu. Ak pre viac názvov filmov vytvoríme rovnakú skratku, zapamätáme si rok natočenia filmu 9999, čo označuje prítomnosť viacnásobných údajov. Samozrejme pokiaľ sú roky natočenia filmu od seba odlišné. Keď budeme mať údaje k filmom načítané, nájdeme všetky súbory s názvom, ktorý nezodpovedá regulárnemu výrazu a pritom obsahuje požadovanú príponu. Pre tieto súbory otestujeme či sa v ich názve nachádza rok filmu, číslo CD, či označenie série a časti seriálu. Ak chýba rok natočenia filmu, môžeme sa po úprave názvu filmu pokúsiť v hashi rok natočenia nájsť.

Ak rok natočenia filmu nájdeme a pokúsime sa uhádnuť jazyk, v ktorom je obsah súboru napísaný, ešte musíme skontrolovať, či upravený názov súboru zodpovedá regulárnemu výrazu. Inak by chyba úplne odstránená nebola. U filmov požadujeme číslo CD, u seriálov označenie série a časti. Ak tieto chyby odstránené nie sú, súbor zmažeme. Môže sa stať, že niektoré údaje sú v názve linku uvedené viackrát. Takéto chyby sa tiež odstránia a použijú sa posledné kontrolované údaje. Veľmi dôležité je, že upravený test dokáže od seba oddeliť a identifikovať jednotlivé údaje. Týmto spôsobom sa podarilo opraviť aj chyby, ktoré vznikli uvedením všetkých požadovaných údajov v názve súboru v nesprávnom poradí. Takto sme zachránili 1168 súborov k filmom a 64 súborov k seriálom. Linky na tieto súbory sa nakopírovali do adresárovej štruktúry, pričom sa vytvorili dva nové adresáre k seriálom.

2.3 Make rough_clean

Pred výberom najlepšieho páru potrebujeme odstrániť súbory, ktoré by mali rovnaký obsah, lebo pri viacnásobnom výskyte by sa výrazne ovplyvnila štatistika a zmenili pravdepodobnosti výskytu jednotlivých viet v textoch. Musíme tiež hľadať súbory, ktoré obsahujú riadky, ktoré sa nebudú prekladať a takéto riadky zo súborov odstrániť. Okrem toho sa môžu v texte nachádzať rôzne značky a chyby, ktoré sa dajú opraviť. V nasledujúcich testoch sa pokúsime tieto problémy vyriešiť.

Pri nasledujúcich testoch budeme potrebovať, aby boli názvy súborov korektné, preto hneď jedným z prvých testov overíme, či sú súbory správne

pomenované. Problematické môžu byť aj preklepy v názvoch súborov. Nie vždy totiž vieme u dvoch súborov s podobným názvom spoľahlivo určiť, či ide o jeden film alebo o dva rozdielne filmy. Pokiaľ však rozdiel spočíva len v umiestnení medzier, vieme takéto chyby odstrániť a názvy súborov zjednotiť. Neskoršie testy sa tak výrazne zjednodušia. Kým však podobné názvy súborov zjednotíme, musíme najprv opraviť všetky chyby v názvoch súborov, ktoré opraviť vieme.

2.3.1 Zlievanie adresárov

Ako prvé sa zlejú adresáre k seriálom, ktoré považujeme za rovnaké. Dva seriály považujeme za rovnaké, ak sa v názve líšia znakmi _ (podtŕžnik), ktoré označujú medzery medzi slovami. Najprv si k menám seriálov nájdeme skratky, ktoré vzniknú odstránením nežiaducich znakov ako sú podtŕžniky, bodkočiarky a hranaté zátvorky. Z nich po zotriedení vyberieme tie, ktoré sa v zotriedenom zozname vyskytujú viackrát. Následne hľadáme adresáre, ku ktorým tieto skratky prislúchajú. Pri hľadaní adresárov, ktorým prislúcha daná skratka, chceme testovať čo najmenej adresárov. Preto využijeme prvé písmeno v skratke a testujeme adresáre, ktoré začínajú na dané písmeno. Toto môžeme urobiť vďaka tomu, že žiaden názov adresára nezačína na podtržník, ktorý by sme pri vytváraní skratky zmazali. Pochopiteľne vzniká otázka koľko znakov zo začiatku názvu seriálu môžeme použiť pri hľadaní pôvodného mena adresáru. Čím viac znakov z pôvodného názvu poznáme, tým menej adresárov musíme testovať. Ako sa ale ukázalo pri seriáloch *A team* a *X files*, ku ktorým sa vytvoria adresáre *a_team* a *x_files*, podtržník môže byť v názve adresára už na druhom mieste. Aby test naozaj odhalil všetky adresáre s podobnými názvami, nemôžeme teda pri hľadaní adresárov použiť viac ako prvý znak. To ale nie je až taká komplikácia nakoľko sa k seriálom dokopy vytvorí len 261 adresárov. Na písmeno *s* pritom začína 66 adresárov, na ostatné menej ako 19 adresárov. Výpočet beží pri takýchto počtoch testovaných adresárov dostatočne rýchlo. Navyše sa do iných adresárov preleje obsah len 16 adresárov.

Ku každému adresáru zistíme, koľko je v ňom súborov a ako sú v ňom súbory pomenované. Počet súborov v adresári nás zaujíma preto, aby sme presúvali čo najmenej súborov medzi adresármi. To dosiahneme tak, že najväčší možný počet súborov ponecháme v svojom pôvodnom adresári. Inak povedané, nájdeme adresár, ktorý má spomedzi zlievaných adresárov v sebe

najviac súborov a do neho presunieme obsah ostatných adresárov.

Vzhľadom na to, že sa pri neskoršom spracovávaní pracuje so súbormi s rovnakým menom (až na unikátne ID), potrebujeme zabezpečiť, aby sa v každom adresári so seriálom nachádzali súbory s rovnakým menom seriálu. Počas vývoja programu totiž vznikali chyby, keď niektoré podprogramy spracovávali jednotlivé adresáre samostatne a keď sa rovnaké mená seriálov v názvoch súborov v jednom adresári nevyžadovali. Iné podprogramy spracovávali celú stromovú štruktúru ako celok. Ak sa v adresári k jednému seriálu nachádzali dva súbory, ktoré boli v rôznych jazykoch a boli si navzájom párom, no neboli rovnako pomenované, pri teste na zhodu mena boli oba považované za súbory, ku ktorým neexistuje pár. Preto musíme názvy súborov v jednom adresári zjednotiť. Tu sa ukáže ďalšia výhoda ponechania adresára s najväčším počtom súborov k danému seriálu. Keď zjednocujeme názvy (prípadne skratky) seriálu v názvoch súborov, premenovávame najmenší možný počet súborov, aký je na zjednotenie potrebný.

2.3.2 Chyby v názvoch súborov

Druhým testom sa kontroluje, či názvy súborov spĺňajú všetky požiadavky. Štruktúru názvu súborov k seriálom i filmom je možné popísať pomocou regulárnych výrazov. Tento test bol výrazne upravený. Pred úpravami sa všetky súbory, ktorých názvy nezodpovedali ani jednému regulárnemu výrazu, odstránili. Po úprave sa tieto súbory ešte ďalej testujú pokiaľ majú jednu z požadovaných prípon.

Tak ako v prípade adresára *problem*, aj pri tomto teste sa snažíme uhádnuť rok natočenia filmu, pokiaľ ho v názve súboru nenájdeme. Opäť využijeme tabuľku s menami a rokmi natočenia filmu, ktorú máme k dispozícii k datam z portálu opensubtitles.org. Aj tu dokážeme z názvu súboru odstrániť duplicitné údaje ako je rok natočenia filmu či označenie série a časti seriálu. V prípade nesprávneho poradia údajov v názve súboru sa údaje zoradia tak, aby upravený názov bol v poriadku. Na rozdiel od testu, ktorý sme použili na súbory v adresári *problem*, vieme, že v adresároch *movies* a *series* sa nachádzajú len súbory s českým alebo anglickým kódom jazyka v názve. Jazyk sa teda nesnažíme uhádnuť, ale si ho zapamätáme.

Pri tomto teste sme premenovali a ušetrili 178 súborov k filmom a 252

súborov k seriálom, ktoré by sa boli predtým automaticky zmazali.

2.3.3 Súbory s podobnými názvami

Problémy pri spracovaní dat vznikali aj vtedy, keď sa mazali súbory s podobnými názvami. Zbytočne sme totiž prichádzali o data, ktoré sme mohli ďalej spracovávať. Preto si vygenerujeme páry s menom súboru a skratkou po zmazaní znakov _ (pre pripomenutie sa do mien filmov dostali nahradením medzier). Postupne tieto páry prechádzame a pre každú skratku zjednotíme mená súborov, z ktorých sme túto skratku dostali. Týmto dosiahneme, že všetky súbory k danému filmu a CD budú mať rovnaký začiatok názvu a tak sa po zotriedení súborov podľa abecedy ocitnú za sebou. Vďaka tomu nám v mnohých prípadoch postačí jednoduché triedenie a ďalšie podprogramy sa budú môcť spoliehať na to, že ak k nejakému súboru neexistuje súbor s rovnakým názvom filmu, potom tento súbor nemá k sebe pár.

Hlavnou výhodou tohto premenovania je, že sa výrazne zjednodušil mazací algoritmus, ktorý maže súbory bez páru v druhom jazyku. Predtým totiž musel zachovávať súbory, ktoré mali podobné meno a kontrola správnosti tohto algoritmu bola zložitá. Teraz je možné zmazať všetky súbory, ku ktorým sa pár hneď nepodarí nájsť. Oproti predošlej verzii projektu nedochádza k zmene algoritmu premenovania súborov s podobnými názvami, ale zmene poradia jednotlivých testov. Kým sa v predošlej verzii najprv mazali súbory s príliš podobnými veľkosťami, v novšej verzii sa najskôr premenujú súbory s podobnými názvami a až nasledovne sa kontrolujú veľkosti súborov. Pri kontrole veľkostí súborov sú totiž presné názvy súborov dôležité a preto je potrebné podobné názvy ešte pred týmto testom zjednotiť.

Pri tomto teste je nájdených 134 názvov filmov, ktoré sú podobné s iným názvom filmu. Všetky súbory s jedným z týchto mien sú následne premenované. Žiadne súbory sa pri tomto teste nemažú. Zabráni sa však mazaniu súborov pri iných testoch, ktoré by nemuseli nájsť pár k niektorému súboru, pokiaľ by sa základ názvu tohto súboru nezjednotil so základom názvu iného súboru.

2.3.4 Súbory, ktoré nemajú pár v druhom jazyku

Keď program skončí, chceme mať na výstupe páry súborov, ktoré sú si navzájom prekladom. Musia byť teda k rovnakému filmu a rovnakému CD, prípadne rovnakej časti toho istého seriálu. Pokiaľ nájdeme súbory, ku ktorým neexistuje súbor k danému filmu či časti seriálu v druhom jazyku, pár súborov s nimi nevytvoríme a tak ich môžeme zmazať. Pri hľadaní nespárovateľných súborov si ukladáme do hashu koľko súborov k danému filmu v danom jazyku sme už našli. Na konci prejdeme mená filmov a seriálov a pokiaľ k danému filmu či seriálu neexistuje aspoň jeden súbor v anglickom jazyku a aspoň jeden súbor v českom jazyku, súbory k tomuto filmu zmažeme. Pokiaľ sa v stromovej štruktúre nachádza nejaký adresár, ktorý je prázdny, zmaže sa. Po skončení mazania ku každému súboru existuje súbor v druhom jazyku, ktorý k nemu môže byť párom.

Vzhľadom na to, že niektoré časti programu pracujú pomerne dlho, je vhodné, aby sme netestovali súbory, ktoré budú neskôr s istotou zmazané. Takýmito súbormi sú aj súbory, ktoré nemajú v druhom jazyku k sebe pár. Zmazanie týchto súborov trvá pomerne krátko a môže prispieť k výraznému zníženiu časovej náročnosti ostatných testov. Preto pred spustením časovo náročných testov zmažeme všetky súbory, o ktorých vieme, že nebudú mať pár v druhom jazyku.

Tento test sme pridali najprv pred test, ktorý v súboroch maže riadky s metainformáciami. Ušetrili sme tak testovanie 15 515 súborov k filmom a 4 981 súborom k seriálom. Pri časovej náročnosti tohto testu môžeme hovoriť o veľkej úspore času. Pritom sa ale ušetrilo len na testoch, ktoré bolo zbytočné vykonávať nakoľko by sa dané súbory aj tak neskôr celé zmazali. Vzhľadom na počet zmazaných súborov ostalo až 122 adresárov k seriálom prázdnych. Tieto sa taktiež zmazali.

Pred testom, ktorý maže súbory s podobnými veľkosťami, bolo mazanie nespárovateľných súborov vykonané už predtým. Zmazalo sa 336 súborov k filmom, 92 súborov k seriálom a 3 adresáre. Pred vyhľadávaním súborov v nežiaducich jazykoch tento test pribudne tiež. Pôvodne sa na tomto mieste hľadali hlásky, ktoré sa často vyskytujú v českých a anglických slovách. Následne pribudol test, ktorý vyhľadáva v súboroch slová, ktoré sa v českom ani anglickom jazyku nevyskytujú. Príliš dlhé zoznamy nájdených riadkov so zakázanými slovami nám spôsobovali problémy pri ich spracovaní. Aby

sa tieto zoznamy skrátili, zmažeme všetky súbory, o ktorých vieme, že ich nezachováme. Pred kontrolou jazyka tak zmažeme 252 súborov k filmom, 34 súborov k seriálom a 10 prázdnych adresárov.

Ďalšie mazanie nespárovateľných súborov sme pridali pred prevod do formátu *plain*. Zasa ide časovo náročnejší výpočet, ktorý nebudeme vykonávať na súboroch, ku ktorým nemôžeme nájsť pár. Preto najprv zmažeme 221 nespárovateľných súborov k filmom, 137 súborov k seriálom a 2 adresáre k seriálom.

Tento test pôvodne napísal pán Zdeněk Žabokrtský. Neskôr prešiel miernymi úpravami podľa toho ako sa menili názvy súborov (po prevode do formátu *plain*, po odstránení čísiel). Počet jeho spustení bol potom ovplyvnení autormi časovo náročných skriptov, ktorí sa rozhodli, že nebudú zbytočne testovať súbory, ktoré sa aj tak zmažú.

2.3.5 Mazanie metainformácií a oprava I a l

Ako už bolo spomínané pri porovnávaní veľkostí súborov, v súboroch sa môžu nachádzať rovnaké texty až na pár riadkov, v ktorých sa nachádzajú informácie o autorovi súboru s titulkami, jeho emailovou adresou, prípadne nejakou URL. Taktiež boli objavené riadky s informáciami o producentovi, filmových štúdiách, prípadne meno osoby, ktorá súbor s titulkami opravovala. Občas sa tiež objavuje v českých titulkoch názov filmu, ktorý sa ale v anglických titulkoch u anglických filmov nemusí objaviť. Našťastie mnohí autori súborov s titulkami si zvykli názov filmu uvádzať v nasledovných tvaroch:

N Á Z O V F I L M U
N Á Z O V F I L M U

Tieto tvary sa dajú pomerne ľahko popísať regulárnym výrazom a tak i ľahko testovať. Pritom druhý tvar býva zapisovaný viacerými spôsobmi. Buď sú tagy <i><b> pred názvom a uzatváracie tagy za ním alebo sú tagy pred každým písmenom a uzatváracie zasa za jednotlivými písmenami.

Nakoľko sa v súboroch budú mazať riadky, bude potrebné celý súbor načítať a nanovo uložiť. Pri mazaní riadkov s nežiaducimi informáciami sa

musia samozrejme zmazať aj riadky s časovými údajmi k nim. V prípade, že sa má zmazať riadok a medzi dvoma časovými údajmi sú s ním aj iné riadky, zmažú sa aj tieto. V prípade, že sa nachádzajú za sebou riadky len s časovými údajmi a žiadnym textom, zmaže sa časový údaj, za ktorým text nenasleduje.

Riadok, ktorý sa má zmazať sa určí tak, že sa v ňom snažíme zmazať zakázané slová, prípadne tvary, ktoré zodpovedajú regulárnym výrazom. Ak dôjde k skráteniu dĺžky riadku, muselo dôjsť aj k zmazaniu niektorej jeho časti. V takom prípade teda riadok musel obsahovať zakázané slovo lebo zakázaný substring ako *www* či *http*.

Nakoľko prepisujeme celý súbor, je vhodné na riadkoch, ktoré vieme, že zachováme vykonať ďalšie testy. Odstraňujú sa tu tagy ¡b¿ , ¡i¿ a ich uzatváracie tagy. Taktiež sa vymažú informácie v hranatých zátvorkách, v ktorých sa nachádzajú informácie o zvukoch a podobne napríklad [klopanie]. Bola sem zahrnutá aj oprava spätných apostrofov a úvodzoviek, ktoré sú následne nahradené riadnymi apostrofmi. Samozrejme k náhrade dochádza len vtedy ak sa zlý apostrof alebo úvodzovky nachádzajú medzi dvoma slovami. Aby sa naozaj ukladanie súborov vykonávalo efektívne a nemuseli sa všetky súbory v inom skripte ešte raz načítavať a prepisovať, pridal sa do tohto skriptu ešte test, ktorý opraví väčšinu l, L, i na I, l na i prípadne I na l a II na ll. Pritom sa zachovávajú všetky rímske číslice v súboroch.

Napríklad riadok
L”m iII.
sa opraví sa
I’m ill.

Takýchto riadkov je v súboroch viacero. Vďaka tomuto testu sa tak poopravujú veľmi ťažko čitateľné a pomerne nepoužiteľné riadky pre štatistický preklad.

Okrem tohto všetkého sa tiež kontrolujú riadky, v ktorých sa nachádzajú texty piesní. Ak je v súbore takýchto riadkov príliš veľa, súbor sa zmaže. Ak je riadkov s piesňami pomenej, zapíšu sa názvy súborov do samostatného logu, keby v budúcnosti bolo treba s týmito súbormi ešte treba pracovať. Kvôli vysokému počtu riadkov s textami piesní bolo zmazaných 2 012

súborov k filmom a 230 súborov k seriálom.

Tento test pôvodne nasledoval až po teste, ktorý maže súbory s podobnými veľkosťami. Test na podobné veľkosti bol vytvorený kvôli tomu, že niektoré súbory sa v datach vyskytovali viackrát a častokrát sa líšili len jedným riadkom. Najprv zmažeme tieto riadky navyše a tak zmenšíme rozdiely medzi veľkosťami podobných súborov. Tým zvýšime efektívnosť nasledujúceho testu.

2.3.6 Súbory s podobnými veľkosťami

Niektoré súbory majú rovnaký obsah a líšia sa len formátom. Prípadne v jednej kópii je meno autora titulkov či preložený názov filmu a v druhej kópii nie. Takto povznikali tisíce súborov, ktoré sa líšia len minimálne, napríklad jedným riadkom s menom prekladateľa. Zvyšok textu v súboroch je totožný. Pri štatistickom preklade nám súbory s podobným obsahom môžu výrazne ovplyvniť pravdepodobnosti výskytov jednotlivých slov, výrazov vo vete za sebou a podobne. Z tohto dôvodu budeme prechádzať adresáre (kvôli rýchlosti triedenia) a v nich zotriedime súbory podľa veľkosti. Ku každému filmu a danému CD, ktoré sme už skontrolovali si budeme pamätať veľkosť posledného kontrolovaného súboru. Ak k tomu istému filmu a danému CD budeme neskôr kontrolovať súbor s veľkosťou väčšou o menej ako 40 Bytov, budeme predpokladať, že väčší súbor obsahuje viac dat ako je potrebné. Data v súbore s menšou veľkosťou teda budeme považovať za kompletné.

U seriálov sme predpokladali veľkosť súborov medzi 20 000 B a 64 000 B a veľkosť súborov k filmom sme akceptovali medzi 15 000 B a 160 000 B. Súbory s veľkosťami nespadajúcimi do týchto intervalov sa mazali. Tu ale vzniká otázka, či tieto medze sú dobre nastavené. Zaujíma nás, ako závisí počet zmazaných súborov mimo zvoleného intervalu na hraniciach tohto intervalu. Okrem toho samozrejme musíme dávať pozor na to, pri akej hranici mažeme súbory, ktoré majú pár a kedy ponechávame súbory, ktoré pár nemajú.

V tabuľke č.2.1 sa nachádzajú údaje o počte súborov k filmom, ktoré sa zmažú ak sa spodná hranica intervalu v algoritme nastaví na hodnotu uvedenú v stĺpci *Limit*. V druhom stĺpci sa nachádzajú počty súborov k filmom, ktoré sa zmažú celkovo. Ide teda o súbory, ktoré nespadajú kvôli svojej

Limit	Celkom	Rozdiel	Pod limit	Rozdiel
15 000	15 851	0	2 748	0
14 000	15 515	336	2 333	415
13 000	15 234	281	2 003	330
12 000	14 969	265	1 692	311
11 000	14 762	207	1 442	250
10 000	14 572	190	1 224	218
9 000	14 450	122	1 081	143
8 000	14 320	130	926	155
7 500	14 247	73	842	84
7 000	14 162	85	739	103
6 000	14 055	107	618	121
5 000	13 950	105	503	115
4 000	13 859	91	397	106
3 000	13 782	77	312	85

Tabuľka 2.1: Dolné limity a počty zmazaných súborov

veľkosti do prípustného intervalu a súbory, ktoré majú podobnú veľkosť ako súbory testované pred nimi. V štvrtom stĺpci sú zasa uvedené počty súborov, ktoré majú menšiu veľkosť ako je spodná hranica intervalu prípustnosti. V treťom a piatom stĺpci sú rozdiely medzi počtami súborov, ktoré sa zmažú pri zmene spodnej hranice na hranicu uvedenú o riadok vyššie. Napríklad pri dolnej hranici 12 000 Bytov sa zmaže 14 969 súborov k filmom, z ktorých 1692 má menšiu veľkosť ako 12 000 Bytov. S veľkosťou od 12 000 do 12 999 Bytov existuje pritom 311 súborov. U 265 z týchto 311 súborov rozhoduje spodná hranica intervalu prípustnosti o tom, či sa súbor zmaže. Zvyšných 46 súborov s veľkosťami od 12 000 do 12 999 Bytov sa zmaže bez ohľadu na veľkosť intervalu prípustnosti. Ich veľkosti sú totiž podobné veľkostiam iných súborov. Na základe nastavenia hraníc intervalu prípustnosti sa algoritmus pri nich rozhoduje, či týchto 46 súborov zmaže pre príliš malú veľkosť alebo pre veľkosť podobnú veľkosti iného súboru k danému filmu a danému CD.

Keď vieme koľko súborov sa pri jednotlivých nastaveniach spodnej hranice intervalu prípustnosti zmaže, potrebujeme overiť kedy sa mažú prevažne súbory, ktoré sa zmazať majú a kedy sa mažú súbory, ktoré by bolo možné spárovať s iným súborom. Necháme si preto vygenerovať zoznamy súborov

s veľkosťami medzi dvoma susednými limitmi a budeme sledovať či sa dané súbory mazať majú alebo nie. Tento test nebudeme robiť automaticky, ale pootvárame si súbory, ktoré sa majú zmazať a súbory k rovnakému filmu, ktoré by k nim mohli byť párom. Overíme, či si naozaj obsahom odpovedajú. Samozrejme kvôli časovej náročnosti tejto kontroly využijeme vzorky 20 až 30 súborov, teda asi 10 až 15% súborov určených na zmazanie.

Pri kontrole súborov s veľkosťami medzi 11 000 a 11 999 Bytov sme zistili, že viac ako 80% súborov sa mazať malo. Preto sme so spodnou hranicou intervalu prípustnosti pod 12 000 Bytov ďalej neuvažovali. Pri kontrole súborov s veľkosťami medzi 12 000 a 12 999 Bytov sme zistili, že z vybranej vzorky bola asi polovica súborov správne určená na zmazanie, štvrtina zmazaná byť nemala a posledná štvrtina súborov mala k sebe obrovské množstvo súborov v druhom jazyku (nad 10), medzi ktorými bolo pravdepodobné, že sa pár nájde. Napriek tomu, že počet súborov určených na zmazanie, ktoré majú byť naozaj zmazané je asi len polovičný, posunieme sodnú hranicu aspoň na 13 000 Bytov. Ak by sme totiž nechali približne 130 súborov, ktoré by boli chybné, výrazne by to znížilo kvalitu korpusu. Ani zachovanie 130 kvalitných súborov by toto zhoršenie kvality nevykompenzovalo. Preto je lepšie nejaké súbory stratiť ako je zachovať súbory, ktoré kvalitu prekladu znižujú.

Pri kontrole súborov s veľkosťami medzi 13 000 a 13 999 Bytov sme zistili, že približne polovica súborov by sa mala zachovať. Zaujímavé ale bolo, že spomedzi súborov určených na zmazanie nemalo približne 80% z nich pár v druhom jazyku. Z tohto dôvodu pred test na kontrolu podobných mien súborov pridáme test, ktorý zmaže súbory bez páru v druhom jazyku. Mazanie súborov bez páru síce nasleduje hneď po mazaní súborov s podobnými veľkosťami (pred mazaním metainformácií), oba tieto testy sú dostatočne časovo a priestorovo náročné na to, aby sa dvojnásobné mazanie súborov bez páru oplatilo vykonať navyše.

Spodnú hranicu intervalu prípustnosti teda nastavíme na 13 000 Bytov. Pre mierne zrýchlenie výpočtu sa miesto volania príkazov *ls* a *rm* zo shellu použijú funkcie *glob* a *unlink* v Perle.

Pri tomto teste ušetríme 453 súborov k filmom a 13 súborov k seriálom, ktoré by sa inak zmazali. Napriek tejto úspore sa zmaže 15 212 súborov k filmom a 6 281 súborov k seriálom. Pred zmenami, ktoré sa vykonali počas

práce na bakalárskej práci, sa zmazalo týmto testom zmazalo len 13 563 súborov k filmom a 4 110 súborov k seriálom. Prečo ale nastala táto zmena keď samotný test po úprave zachováva väčší počet súborov? Dopomôcť tomu mohla aj zámena poradia testov medzi sebou. Teraz sa zmažú riadky s metainformáciami, a až potom sa zmažú súbory s podobnými veľkosťami. Ak sa dva súbory predtým líšili pár riadkami s metainformáciami, teraz majú podobnú veľkosť a sú zmazané.

2.3.7 Kontrola používania interpunkcie

Pri párovaní viet v súboroch v najlepšom páre je problémom ak sa nie je jasne rozpoznateľné čo je a čo nie je veta. Inak povedané, ak v texte chýbajú interpunkčné znamienka, nemusíme vždy vedieť určiť hranicu vety. Z tohto dôvodu sa zmažú súbory, v ktorých sa interpunkcia nevyužíva alebo sa využíva málo.

Načítame 100 riadkov s textom. Ak narazíme na riadok, ktorý bude obsahovať len časové údaje, načítame miesto neho ďalší riadok. Keď budeme mať načítanú vzorku, spočítame koľko otáznikov, výkričníkov, bodiek a čiarok sa vo vzorke nachádza. Ak sme nenašli aspoň 25 interpunkčných znamienok, potom súbor zmažeme. Ak sa bude v súbore nachádzať aspoň 50 interpunkčných znamienok, potom súbor zachováme bez zmeny. Inak zapíšeme meno súboru do logu a pustíme na neho opravný skript.

V adresárovej štruktúre sme zmazali 843 súborov k filmom a 44 súborov k seriálom. Pritom sme mazali súbory, v ktorých nebolo možné interpunkciu naozaj kvalitne opraviť a súbory, ku ktorým nám ešte ostávala náhrada v podobe iného súboru k tomu istému filmu či seriálu.

2.3.8 Oprava interpunkcie

Ak je v súbore možné doplniť interpunkciu, pokúsime sa o to. Pri úprave interpunkcie v danom riadku bude pritom dôležité poznať riadok aktuálny aj riadok nasledovný. Napríklad na koniec riadku nemôžeme dať bodku, pokiaľ ďalší riadok začína malým písmenom. Pravdepodobne totiž ide o pokračovanie tej istej vety, ktorú nesmieme ako celok rozdeliť. Ak sa napríklad na nasledujúcom riadku nachádza ako prvé slovo spojka *že*, potom musíme skontrolovať, či sa na konci aktuálneho riadku nachádza čiarka. Taktiež

ukončíme vetu pokiaľ narazíme na nový časový údaj. To znamená, že budeme teda predpokladať, že sa celá veta na obrazovke zobrazí naraz.

Testo test súbory nemaže, ale prepisuje. Interpunkcia sa tak opraví v 86 súboroch k filmom a v 3 súboroch k seriálom.

2.3.9 Mazanie súborov v iných jazykoch

Je nežiaduce, aby sa do najlepšieho páru dostal súbor v inom jazyku ako českom a anglickom. Z tohto dôvodu je preto potrebné súbory v iných jazykoch čo najskôr odstrániť. V pôvodnom teste pána Žabokrtského sa kontrolovalo koľko hlások charakteristických pre anglický prípadne český jazyk sa v súbore nachádza. V prípade, že bol počet hlások príliš malý, došlo k zmazaniu súboru.

K tomuto testu pribudol nový test. V adresárovej štruktúre sa hľadajú v súboroch riadky, ktoré obsahujú slová, ktoré sa v anglickom ani českom jazyku nevyskytujú. Medzi súbormi sme našli aj také, ktorých obsah je v slovenskom, francúzskom, nemeckom, maďarskom a španielskom jazyku. Pre tieto jazyky sme vybrali slová *som*, *la*, *ein*, *nem* a *con*. Pokiaľ v nejakom súbore nájdeme aspoň 15 výskytov niektorého z týchto slov, potom text v súbore považujeme za nevhodný pre výber do najlepšieho páru a súbor zmažeme.

Na hľadanie výskytov zakázaných slov sa využíva príkaz *grep*. Ten ale nie je možné spustiť naraz pre všetky súbory, nakoľko sa na zozname súborov vypisujú aj vety, v ktorých zakázané slovo bolo nájdené. Tento zoznam je preto príliš dlhý. Kvôli tomu sa musia prechádzať adresáre postupne. V každom adresári prechádzame možné dvojice prvých dvoch písmen. Zakázané slová potom hľadáme v súboroch začínajúcich na aktuálnu dvojicu písmen. Samozrejme, že zakázané slová hľadáme len v prípade, ak nájdeme nejaké súbory, ktoré na aktuaálnu dvojicu písmen reálne začínajú. Inak by sme test spúšťali zbytočne. Pre úplnosť ako druhé písmeno, či správnejšie druhý znak v názve súboru akceptujeme aj podtŕžník.

Pri tomto teste sa nám podarilo nájsť odstrániť 591 súborov k filmom a 36 súborov k seriálom, ktorých obsahy by mali v slovenskom, francúzskom, nemeckom, maďarskom alebo španielskom jazyku.

2.3.10 Test na chýbajúcu diakritiku

Nasledujúcim testom sa pokúsime odhaliť súbory, ktoré by podľa názvu mali byť v českom jazyku, no ich obsah v tomto jazyku napriek tomu napísaný nie je. U českých súborov kontrolujeme, či sa v súbore používa diakritika. Pritom predpokladáme, že v súbore napísanom v českom jazyku sa na prvých 200 riadkoch nachádza slovo, v ktorom sa vyskytuje samohláska s dĺžňom alebo spoluhláska s mäkčeňom. Ak ale nájdeme súbor, v ktorom sa diakritika nepoužíva, zmeníme príponu súboru na *.del* a neskôr ho zmažeme. Tým, že sme časť súborov premenovali, a tak ich pripravili na zmazanie, stratili iné súbory k sebe pár. Preto ich nemá zmysel ďalej spracovávať a môžeme ich zmazať.

2.4 Make toplain - prevod súborov do formátu plain

Vzhľadom na to, že máme k dispozícii súbory v rôznych formátoch napísané stovkami autorov, je potrebné obsah súborov previesť do spoločného formátu, aby sme s nimi mohli ďalej pracovať. Obsah súborov prevedieme do formátu *plain*. Ten bude obsahovať iba text, ktorý nás zaujíma pri štatistickom preklade. V súboroch zmažeme všetky zátvorky aj s ich obsahom, číselné údaje oddelené od seba bodkočiarkami, čiarkami a inými oddeľovačmi. Tieto číselné údaje udávajú časy, kedy má byť daný text zobrazený. Prázdne riadky, ktoré sa môžu v niektorých formátoch vyskytovať, by už mali byť odstránené testom, ktorý maže riadky s metainformáciami. Niektoré riadky ale obsahujú iba časové údaje. Aby nám neostali ďalšie prázdne riadky, bude treba riadky obsahujúce časové údaje celé zmazať.

Tento test napísal pán Zdeněk Žabokrtský. Neskôr došlo k ešte jednej miernej úprave nakoľko sa v teste pôvodne rozoznávali tri formáty súborov. Na základe zisteného formátu sa uskutočnil prevod obsahu do formátu *plain*. Test na prítomnosť metainformácii zmazal zo súborov prázdne riadky a riadky s celými číslami, čím sa zjednotili dva formáty a teda sa aj upravil tento test.

2.4.1 Nežiaduce znaky v texte

Niekedy je text zobrazovaný v jednom momente príliš dlhý na to, aby sa dal zobraziť na monitore v jednom riadku. Z toho dôvodu sa pochopiteľne text zobrazí vo viacerých riadkoch. Nie práve najlepším zvykom niektorých autorov je používať v takýchto prípadoch tri bodky na koncoch a začiatkoch po sebe nasledujúcich riadkov. Tieto bodky síce naznačujú, že text na týchto riadkoch patrí k sebe a je hovorený tou istou osobou, pri preklade nám však budú prekážať. Preto sa skupiny aspoň troch bodiek zmažú. Pritom sme museli ale dávať pozor, aby sme pri predošlých testoch nezmenšili tieto skupiny bodiek. Viackrát sa totiž dvojnásobné medzery, bodky a čiarky mazali. Ak by sme však skupiny troch bodiek zredukovali na jednu bodku na konci riadku a na začiatku ďalšieho riadku, dostali by sme bodku na mieste, kde veta nekončí a na začiatku riadku, kde veta bodkou nikdy nezačína. Preto bolo potrebné vždy dávať pozor na to, aby sa skupiny aspoň troch bodiek zachovali a pri prevode do formátu *plain* riadne odstránili.

Taktiež sa na začiatkoch niektorých riadkov môžu nachádzať pomlčky. Tie sú väčšinou informáciou o tom, že text na tomto riadku je hovorený inou osobou ako text z riadku predchádzajúceho. Aj keď by sa zdalo, že tieto údaje môžu byť pri párovaní užitočné, v datach sa pomlčky vyskytujú výnimočne a nemáme zaručené, že sa v oboch súboroch k danému filmu a danému CD budú nachádzať. Testovanie na prítomnosť týchto znakov by nebolo efektívne a preto ho vynecháme a tieto znaky z textu odstránime.

2.4.2 Strata údajov o čase zobrazenia

V tomto momente sme odstránili číselné časové údaje, ktoré prekladať nikdy nebudeme. Zároveň sme však stratili údaje, kedy sa ktorý text zobrazí. Tieto údaje nám mohli pomôcť pri párovaní textov nakoľko predpokladáme, že vety, ktoré sú si navzájom prekladom, sa zobrazia približne v rovnakom čase (s malou toleranciou rozdielu). Problematické by však boli porovnávania, ak by sme v jednom súbore mali časový údaj v minútach a sekundách a v druhom by bol reprezentovaný číslami framov, kedy sa má text zobraziť. U väčšiny filmov je možné počet framov prepočítať na ekvivalentný čas. Pokiaľ by sa framy v niektorom filme vyskytovali častejšie ako 24 či 25-krát za sekundu, potom by sa naše prepočty výrazne skreslili a párovali by sme ku sebe vety, ktoré si nie sú navzájom prekladom.

To, čo by však mohlo byť záchytným bodom v oboch prípadoch, sú väčšie medzery. Napríklad, počas akčnej scény nie je dôležité čo herci hovoria, ale dôležité sú filmové efekty, a tak herci niekedy celé minúty ani nehovoria. Ak by bol v dvoch súboroch mierny posun napríklad o pár sekúnd, na základe času by sme jednoznačne nemohli k sebe priradiť dve vety. V prípade minútovej pauzy by sme však vedeli text rozdeliť na dve menšie časti (ako paragrafy v knihe). Existencia pauzy by v súboroch mohla byť reprezentovaná špeciálnou postupnosťou znakov. Ak by sa pauza nachádzala v oboch súboroch, vedeli by sme, že texty pred pauzou patria k sebe a vety za pauzou patria k sebe. Súčasne by nebolo možné hľadať preklad vety, ktorá bola v prvom súbore pred pauzou, medzi vetami v druhom súbore umiestnenými za pauzou.

Prax ale ukázala, že sa pauza nemusí v oboch súboroch v rovnakom čase vyskytovať. Niektoré filmy, a najmä seriály, začínajú rekapituláciou predošlých častí, úvodnými scénami, a až po nich nasleduje zvučka. V tejto pasáži sa môže objaviť text, ktorý bude preložený len do jedného jazyka. Kým v jednom súbore vznikne asi minútová medzera, v druhom vzniknú dve menšie. Takto by sme v jednom súbore označili špeciálnou postupnosťou znakov pauzu, ktorá by v druhom súbore neexistovala. Z tohto dôvodu by sme okrem špeciálnej postupnosti znakov potrebovali aj časový údaj (stačila by minúta). Ten by sa dal približne vypočítať aj v prípade, keď sa v datach používajú čísla framov. Problém by však mohol nastať, ak by sme spracovávali film vo vyššej kvalite a framy by sa v ňom menili častejšie. V takomto prípade by bol vypočítaný časový údaj skreslený.

Nakoniec sa od označovania dlhších medzier upustilo keďže nemáme istotu, že ich nájdeme v oboch súboroch v najlepšom páre.

2.4.3 Počet súborov vo formáte plain

Pred prevodom do formátu *plain* bolo v adresárovej štruktúre zachovaných 38 499 súborov k filmom a 7 184 súborov k seriálom. Tieto sa nachádzali v 108 adresároch. Po prevode do formátu *plain* sa v adresárovej štruktúre nachádzalo 76 320 súborov k filmom. To znamená, že 37 821 súborov k filmom sa do formátu *plain* previedlo a 678 súborov sa previesť nepodarilo. Súborov k seriálom sa do formátu *plain* previedlo 7 072 a 122 súborov sa previesť nepodarilo.

2.5 Make rename00 – premenovanie súborov k seriálom

Pri súboroch k seriálom nám robia vážny problém chýbajúce čísla sérií. Samotné čísla častí vrámci série taktiež nie sú vždy správne. Pri premenovávaní súborov dôjde k zmazaniu týchto čísiel keďže ich nemôžeme považovať za dôveryhodné. Aby sme ale zachovali informáciu, že ide o seriál, necháme v názve súboru 0x0.

Ako sa neskôr zistilo, v niektorých súboroch sa v texte nachádza informácia o tom, ku ktorej časti a ktorej sérii daný súbor patrí. Preto pribudol po prevode do formátu *plain* test, ktorý v súboroch k seriálom vyhľadá informácie o časti a sérii a súbory následne premenuje. V 922 z 7 072 súborov k seriálom sa túto informáciu podarilo nájsť a súbory následne premenovať.

2.6 Make del_dupl – zmazanie súborov k seriálom s rovnakými veľkosťami

Tento test je podobný testu clean_velkosti.pl. Opäť vychádzame z toho, že súbory s rovnakým obsahom majú rovnakú veľkosť. Nakoľko boli obsahy súborov prevedené do formátu *plain*, už nám neprekáža ak boli pôvodné súbory vo formátoch srt a sub. Teraz sú formáty jednotné a obsahy zhodné. Môžeme tak vykonať lepší test na zhodu obsahov. Budeme totiž porovnávať veľkosti reálnych dat, ktoré sa zobrazujú. Predtým stačila malá zmena vo formáte a veľkosti dvoch obsahovo podobných súborov sa výraznejšie líšili.

Vzhľadom na to, že metainformácie boli zmazané, môžeme testovať súbory na zhodu veľkostí. Predtým sme kvôli potenciálnym riadkom s metainformáciami mazali súbory, ktorých veľkosti sa líšili o menej ako 40 Bytov. Teraz by sa už metainformácie v súboroch nemali nachádzať a tak by mali byť veľkosti väčšiny podobných súborov zhodné.

Medzi súbormi k seriálom sa nám podarilo nájsť 1064 párov s rovnakou veľkosťou. Jeden súbor v každom páre sa premenuje tak, že bude mať na konci príponu *.del*.

2.7 Make select_pairs – výber najlepšieho páru

Pôvodne sa najlepšie páry vyberali úplne iným testom. Ten pracoval na základe toho, že si ku každému filmu a každému CD našli všetky súbory v oboch jazykoch. Rozoznávali sa 4 prípady podľa počtu súborov v jednotlivých jazykoch: 1-1, 1-N, M-1 a M-N. V prípade 1-1 sme pre pár súborov vypočítali takzvaný index zhody a pokiaľ bol dosatočne vysoký, tento pár sme zachovali, inak sme ho zmazali. V prípadoch 1-N a M-1 sme vytvorili N a M indexov zhody a zachovali sme pár s najvyšším indexom zhody, samozrejme len za predpokladu, že bol vyšší ako minimum požadované v prípade 1-1. V prípade M-N sa porovnalo M a N. V menšej množine súborov sa našiel ten, ktorý bol pre danú množinu charakteristický. To znamená, že pokiaľ sme v množine súborov našli pre viaceré súbory nejakú spoločnú vlastnosť, chceli sme, aby zostala zachovaná aj v súbore v najlepšom páre. Preto sme znova vytvorili index, ktorý hovoril o tom, koľko často vyskytujúcich sa slov či častí slov sa v danom súbore nachádza. Do najlepšieho páru sme potom vybrali súbor s najvyšším indexom a ďalej sme postupovali ako v jednom z prípadov 1-N a M-1.

Pri vytváraní indexu zhody sme v prvých 100 riadkoch oboch súborov hľadali rovnaké slová začínajúce veľkými písmenami, prípadne slová, ktoré začínajú veľkým písmenom a zhodujú sa v nich aspoň prvé 3 písmena. Tieto slová by mohli byť totiž menami postáv, prípadne miest, ktoré by sa mohli nachádzať v oboch súboroch len málo pozmenené. Niektoré mená sa totiž neprekladajú vôbec, prípadne sa len skloňujú. Do úvahy sme tiež brali počet otáznikov a výkričníkov v oboch súboroch. Zaujímala nás aj dĺžka viet, ktoré by si mali byť párom. Ak boli dĺžky aspoň polovice testovaných viet podobné, potom sme navýšili index zhody.

Tento test u súborov k filmom pracoval pomerne dobre. Mazal prevažne páry, ktoré boli chybné, zachovávalo sa pomerne veľa párov, ktoré si boli prekladom. Niekedy sa však index zhody napočítal dostatočne vysoký aj u párov, ktoré si neboli prekladom. Najhoršia ale bola malá úspešnosť pri párovaní súborov so seriálmi. Čísla sérií neboli k dispozícii a ako sa neskôr ukázalo, ani čísla častí neboli vždy správne. Keď sa čísla častí a sérií zmenili do tvaru 0x0, dokázal tento test zo všetkých súborov vybrať len jeden najlepší pár. Jednou z možností bolo zväčšiť počet riadkov, ktoré sa načítajú a tak zvýšiť hodnovernosť indexu zhody. Súčasne ale bolo potrebné, aby test

dokázal vyberať viac párov. Pôvodná idea bola nájsť najlepší pár, zmazať podobné súbory, vo zvyšných súboroch nájsť ďalší pár a takto postupovať kým sa nespárujú alebo nezahodia všetky súbory. Nakoniec sa rozhodlo, že sa najlepšie páry budú vyberať pomocou nového testu pána Václava Nováka.

Nový test pracuje na podobnom príncipe, ale efektívnejšie. Najprv sa totiž vyberajú množiny súborov s podobným obsahom. Potom sa medzi jednotlivými množinami hľadajú tie, ktoré k sebe patria a vrámci týchto dvoch množín sa povyberajú súbory do najlepších párov. Pritom sa načítavajú väčšie vzorky, ktoré zaručujú vyššiu kvalitu testovania. Tento nový test bol vytvorený najprv kvôli spárovaniu súborov k seriálom, neskôr sa použil aj na súbory k filmom. Pôvodný test, ktorý nie je až taký efektívny, bol vynechaný, nakoľko nemalo zmysel oba testy zachovať.

Nový test pána Nováka je pomerne časovo náročný a nie je možné ho často opakovať. Doposiaľ sa efektívnosť jednotlivých testov a ich úprav porovnávala podľa počtu súborov, ktoré sa zmazali, prípadne ušetrili. Doposiaľ boli počty súborov napočítané na stroji autora bakalárskej práce Petra Beňu. Ďalšie údaje o počtoch zachovaných súborov pochádzajú z logov vygenerovaných testami na stroji pána Nováka. Testy na oboch strojoch ale nezbehli v jeden deň a preto sú počty zachovalých súborov na oboch strojoch v tejto fáze spracovania odlišné.

Krok	**Movies**	**Series**	**del M**	**del S**	**plain M**	**plain S**
P. Beňa	76 320	14 256	372	1 095	0	0
V. Novák	102 601	25 971	42 634	11 265	8 965	3 278

Tabuľka 2.2: Počet súborov na strojoch P. Beňu a V. Nováka

Ako je vidno z tabuľky č.2.2, pán Novák mal k dispozícii oveľa viac súborov. To vyplýva z toho, že pri párovaní samotnom sa súbory tvoriť nemali, teda už pred výberom najlepšieho páru ich malo byť viac. To nám potvrdzuje aj 23 197 súboroch nájdených vo formáte *srt*, 38 646 súborov vo formáte *sub* a 1 362 vo formáte *txt*, ktoré sme v adresárovej štruktúre našli. Pre pripomenutie, na stroji Petra Beňu sa pred prevodom do formátu *plain* nachádzalo v pôvodných formátoch len 38 499 súborov k filmom a 7 184 súborov k seriálom.

Na stroji pána Nováka sa vytvorilo 5 021 párov k filmom a 2 520 párov k seriálom.

2.7.1 Odstránenie ID z názvov súborov

Keď už vieme, že ku každému filmu a jeho CD máme v českom aj anglickom jazyku maximálne jeden súbor, nepotrebujeme unikátne ID v názve súboru na ich odlíšenie. Súbory k seriálom sa premenovali tiež tak, aby bolo možné z názvov súborov ID odstrániť. Meno filmu, rok, číslo CD a jazyk nám preto postačia na jednoznačnú identifikáciu súboru spomedzi všetkých súborov, ktoré sme zachovali v adresárovej štruktúre. Aby sa mená súborov skrátili, rozhodlo sa, že sa ID pôvodných súborov zmažú (pre pripomenutie pracujeme v adresárovej štruktúre so symbolickými linkami). Ak máme súbor k v nejakom jazyku, ID pôvodného súboru k súboru v najlepšom páre je ťažké uhádnuť. Po zmazaní ID stačí zmeniť v názve súboru len kód jazyka na to, aby sme zistili názov druhého súboru v najlepšom páre.

Tu ešte nastala malá úprava, nakoľko sa premenovávali aj súbory v iných formátoch ako *plain*. Táto chyba bola opravená. Ušetrilo sa tak 116 395 premenovaní. Zaujímavé ale je, že sa podľa logu premenovalo len 8 965 súborov k filmom a 3 278 súborov k seriálom. Podľa predošlého logu malo byť vytvorených 5 021 párov k filmom a 2 520 párov k seriálom.

2.8 Make clean_after_pairing

Ako už bolo spomínané, väčšinu súborov, ktoré nechceme, aby sa dostali do najlepšieho páru, sme sa pokúsili vymazať ešte pred ich výberom do najlepšieho páru. Niektoré testy ale môžeme vykonať len po prevode do formátu *plain*, prípadne na čo najmenšom počte súborov kvôli ich časovej zložitosti.

Ako prvá po premenovaní súborov nasleduje kontrola, či majú naozaj všetky súbory k sebe pár v druhom jazyku. Niektoré súbory sa totiž do formátu *plain* nemuseli previesť. Bolo nájdených 66 súborov k seriálom, ktoré nemajú k sebe pár. Súbory k filmom bez páru v druhom jazyku sa nenašli.

2.8.1 Zmazanie párov s pomerom veľkostí mimo toleranciu

Pri nasledujúcom teste budeme zohľadňovať veľkosti súborov. Pri dvoch testoch sme už síce veľkosti súborov zohľadňovali, vtedy nás však zaujímalo, či dva súbory k tomu istému filmu a tomu istému CD nemajú podobnú veľkosť natoľko, aby sme mohli predpokladať, že ich obsah je takmer identický. Teraz budeme kontrolovať, či majú súbory v páre k danému filmu či časti seriálu podobnú veľkosť. Budeme predpokladať, že preklady nebudú mať rovnakú veľkosť a zároveň rozdiel nebude príliš veľký. Pripustíme pomer veľkostí súborov medzi 4:5 a 5:4. Vychádzame totiž z predpokladu, že jednotlivé vety, ktoré sú si prekladom, sa budú líšiť maximálne o jedno, prípadne dve slová. Ak by sme mali dve vety, z ktorých jedna by bola niekoľkonásobne dlhšia ako prvá, zrejme by nešlo o vety s rovnakým významom. Preto ak nájdeme súbory, ktoré sa aj vo formáte *plain* budú veľkosťami líšiť veľmi výrazne, budeme predpokladať, že si nie sú kvalitným a úplným prekladom.

Mohlo by sa pritom zdať, že tento test je na nesprávnom mieste a mal byť vykonaný skôr. Ako sa však ukázalo, bolo nevyhnutné odstrániť časové údaje a data, ktoré sa prekladať nebudú. Po ich odstránení sa zmenšili veľkosti väčšiny súborov o 20 až 40 percent, vo výnimočných prípadoch až o 60%. Pritom sa nájdu páry súborov s podobnou veľkosťou vo formáte *plain*, ktorých pomer veľkostí v pôvodných formátoch sa blížil k 2. Ak by sme ale taký veľký pomer predtým pripustili, bol by test neúčinný, keďže by skoro všetky páry podmienky testu spĺňali.

Test našiel 240 filmových párov a 27 párov k seriálom, v ktorých pomer veľkostí súborov bol neprípustný. Ku všetkým súborom v týchto pároch je pridaná prípona *.del* a sú určené na zmazanie. Autorom testu je pán Zdeněk Žabokrtský.

2.8.2 Prísnejší test na prítomnosť podobných slov

Tento test pána Žabokrtského je podobný testu pána Beňu, ktorý sa pôvodne vykonával pri hľadaní najlepšieho páru súborov, ktoré by si boli najkvalitnejším prekladom. Kontroluje sa v ňom, či sa v oboch súboroch nachádzajú rovnakí hrdinovia, a to tak, že sa v textoch hľadajú zhodné mená alebo aspoň začiatky mien. Testuje sa zhoda prvých 4 písmen v slovách,

ktoré sa nachádzajú vo vzorke, ktorá má 8 000 znakov. Prvých 10 riadkov sa neberie do úvahy, keby sa tam náhodou nachádzalo meno filmu. Aj keď pribudol test, ktorý zo súborov maže riadky s metainformáciami, meno filmu nemusí byť ľahko odhaliteľné. Ponechaniu páru len na základe zhody v mene filmu sa chceme vyhnúť. Mohlo by sa totiž stať, že budeme mať dva rôzne filmy, ktoré budú mať v názve rovnaké krstné meno hrdinu, no nepôjde o tú istú osobu. Prípadne jeden film bude pokračovaním druhého. Na ponechanie súborov vyžadujeme aspoň jednu zhodu. Ak neberieme do úvahy prvých 10 riadkov, táto zhoda by nemala vzniknúť v názve filmu.

Ako už vieme, pôvodný výber najlepšieho páru bol nahradený novým skriptom. Predpokladalo sa, že tento test už bude neúčinný, nakoľko sa pri výbere najlepšieho páru splní podmienka pre zachovanie páru týmto testom. Napriek tomu as zmazalo 19 párov k filmom a 30 párom k seriálom.

2.8.3 Oprava preklepov pomocou spellu a aspellu

Tak ako asi v každom texte, aj v súboroch s titulkami sa vyskytujú preklepy. Tie môžu samozrejme skomplikovať štatistický preklad, keďže slovo či celú vetu ťažko nájdeme v texte pokiaľ nie je napísaná správne. Aby sme aspoň z časti zmiernili tento problém, pokúsime sa preklepy opravovať. Pritom použijeme príkazy *spell* a *aspell*.

Ako prvé sa postupne načítajú riadky zo všetkých súborov a z týchto riadkov získame jednotlivé slová. Pri každom výskyte slova pripočítame v hashi jednotku k číslu určujúcemu počet výskytov daného slova v datach. Tieto slová sa uložia kvôli kontrole do logu. Veľký význam tohto logu sa ukázal pri hľadaní súborov v nežiaducich jazykoch. U vybratých slov, ktoré mali potenciál vyskytovať sa v súboroch v danom jazyku najčastejšie, sme ľahko zistili počet ich výskytov. Vyberali sa pritom slová, ktoré sa určite nevyskytujú v českom ani anglickom jazyku. Do testu sme z týchto slov vybrali slovo, ktoré sa vyskytovalo v datach najčastejšie. Tak bola pravdepodobnosť odhalenia nesprávneho jazyka daným testom najvyššia.

Následne sa začnú druhýkrát prechádzať jednotlivé súbory a pomocou príkazu *spell* sa získajú slová, ktoré nie sú spisovné v anglickom jazyku. Pokiaľ je týchto slov viac ako 200, je malá pravdepodobnosť opraviť toľko typov chýb správne a súbor radšej zahodíme. Je potrebné si uvedomiť, že

nejde len o 200 slov v celom súbore, ale 200 rôznych slov, z ktorých každé sa môže v súbore vyskytovať viackrát. Ako sa ukázalo, pri 500 chybných slovách už väčšinou ide o text v nežiaducom jazyku.

V prípade, že súbor zachováme, pokúsime sa poopravovať jednotlivé chyby. Pokiaľ sa dané slovo vyskytuje v celých datach menej ako 10-krát, potom sa pomocou príkazu *aspell* pokúsime nájsť k nemu náhradu. Dôvodom, prečo netestujeme aj slová s častejším výskytom je ten, že mená postáv či miest nemusia byť všetky v slovníku, ktorý využíva príkaz spell. Je nepravdepodobné, že by v danom slovníku boli krstné mená a priezviská všetkých postáv, aké sa kedy vo filmoch a seriáloch objavili. Budeme teda testovať slová, ktoré sa v celých datach vyskytujú výnimočne. Pritom sa najprv pozrieme do hashu, či sme dané slovo už testovali a pokiaľ áno, načítame z hashu jeho náhradu. Ak sme ho netestovali, otestujeme ho pomocou príkazu *aspell*. Žiaľ tento príkaz musíme spúšťať z pohľadu časovej náročnosti veľmi nevhodným spôsobom, nakoľko nevieme zabezpečiť interaktivitu pri testovaní a musíme spúšťať ďalšie procesy. Aby sme test urýchlili, snažíme sa pri každej možnej príležitosti vyhnúť zbytočnému testovaniu pomocou tohto nástroja. Preto sa aj všetky už otestované slová ukladajú do hashu aj s ich náhradami, nakoľko z premenných vieme získať informácie veľmi rýchlo.

Problematické je tiež to, že príklad *spell* kontroluje len slová, ktoré sa nevyskytujú v anglickom jazyku. Aby sme tento test mohli rozšíriť aj na súbory, ktoré by mali byť v českom jazyku, bude treba vytvoriť slovník s akceptovanými slovami, s ktorým by príkaz *spell* dokázal pracovať.

Údaje o tom, koľko súborov bolo zmazaných týmto testom, nemáme k dispozícii. Po napočítaní štatistík pánom Novákom došlo totiž k miernej optimalizácii práve tým, že sa zaviedlo mazanie súborov s veľkým počtom chybných slov. Verzia, ktorú mal k dispozícii pán Novák súbory ešte nemazala. Ako sa ukázalo, ani nasledovný test nič nezmazal, nakoľko predošlé testy určovali celé páry na zmazanie, nie jednotlivé súbory. Test mažúci súbory bez páru v druhom jazyku zmaže maximálne toľko súborov koľko zmaže test opravujúci preklepy.

2.9 Make finalclean – prečistenie súborov na konci

Na konci testovania nám okrem súborov vo formáte *plain* ostanú v adresárovej štruktúre aj súbory, ktorým sme príponu upravili na *.del*. Tieto súbory označené na zmazanie v tomto okamihu zmažeme. Okrem toho sa v adresárovej štruktúre môžu nachádzať súbory v pôvodných formátoch, nakoľko sme ich po prevode do formátu *plain* nemazali. Po premazaní v adresároch ostanú len súbory v nami požadovanom formáte.

Pri mazaní všetkých súborov iných ako vo formáte *plain* predpokladáme, že sa nám v adresárovej štruktúre neželané súbory naozaj nenachádzajú. Prekvapujúce ale je, že sa podľa logy zmazalo len 24 438 súborov, pričom sme mali vyše 53 000 súborov určených na zmazanie. Ako môžeme vidieť v tabuľke č.2.3, počty zmazaných súborov sú najozaj nízke a nenapĺňajú naše očakávania.

Typ	***.del***	***.srt***	***.sub***	***.txt***	**Spolu**
skutocnost M	9 058	5 430	7 191	293	21 529
skutocnost S	1 792	818	321	7	2 921
predpoklad M	42 372	17 326	32 794	1 151	93 643
predpoklad S	11 265	5 762	5 472	194	22 693

Tabuľka 2.3: Počty zmazaných súborov

Po zmazaní neželaných súborov nám mohli v adresárovej štruktúre ostať prázdne adresáre. Najmä u seriálov je po zmazaní viacerých súborov vyššia pravdepodobnosť, že adresár ostane prázdny. Z tohto dôvodu prázdne adresáre nájdeme a zmažeme. Dokopy sa zmaže 18 adresárov.

Týmito dvoma mazaniami sme získali istotu, že v adresárovej štruktúre máme len adresáre, ktoré majú neprázdny obsah a že všetky súbory v nich sú vo formáte *plain*.

Kapitola 3

Možnosti na ďalšie rozšírenie projektu

Bakalárska práca nadväzuje na ročníkový projekt autora Petra Beňu. Na konci dokumentácie k tomuto projektu sa navrhovala úprava výberu najlepšieho páru, ktorá bola vykonaná. Miesto zmeny parametrov pôvodného algoritmu sa však nahradil celý skript novým skriptom pána Václava Nováka, ktorý výrazne zlepšil kvalitu párovania súborov k seriálom. Druhým navrhovaným vylepšením projektu bolo spracovanie adresáru *problem*, v ktorom sa nachádzalo 8 340 súborov, ktoré sa pri párovaní nevyužili. V aktuálnej verzii programu sa súbory v adresáre *problem* po testoch zaradia do adresárovej štruktúry a ďalej spracovávajú. Tretí návrhom bolo využiť časové údaje vyskytujúce sa v obsahoch súborov na to, aby sme dokazali lepšie párovať jednotlivé vety k sebe. Tieto údaje sa však napokon nevyužili. Okrem toho sa dodatočne objavili ďalšie chyby. Niektoré súbory v iných jazykoch prešli celou sadou testov, v súboroch sa našli ianformácie, ktoré trebalo zmazať, prípadne niektoré znaky nahradiť. Skoro všetky z týchto problémov boli odstránené.

Priestor na vylepšenie ešte poskytuje test, ktorý hľadá v súboroch slová, ktoré by nemali patriť do jazyku, ktorý je deklarovaný v názve súboru. Pokiaľ je to možné, preklepy v týchto slovách aj odstráni. Pri hľadaní neznámych slov a slov s preklepmi sa používajú príkazy *spell* a *aspell*. Príkaz *spell* však dokáže hľadať len slová, ktoré sa nevyskytujú v anglickom jazyku. Aby sa dali testovať súbory písané v českom jazyky, možno by bolo vhodné použiť iný príkaz. Vďaka príkazu *spell* sa ale dá rýchlo zistiť počet chybných slov

v celom súbore a ušetriť tak veľa ďalších testovaní ak sa rozhodneme daný súbor zmazať. Preto je vhodné ho naďalej využívať. Samozrejme by tu bola ešte možnosť vytvoriť slovník pre príkaz *spell*, v ktorom by boli všetky povolené slová v súboroch v českom jazyku uvedené tak, aby s týmto slovníkom dokázal príkaz *spell* pracovať. V prípade existencie českých slovníkov pre príkazy *spell* a *aspell* by sa potom dal test spustiť aj na súbory v českom jazyku.

Kapitola 4

Záver

Hlavným cieľom bakalárskej práce bolo zvýšiť kvalitu časti paralelného korpusu Czeng, ktorá bola vytvorená z anglických a českých titulkov k filmom a seriálom. Tieto súbory majú tisíce autorov a preto je ich prečistenie náročné, nakoľko takýto počet autorov nedodržiava tie isté pravidlá. Napriek tomu môžeme prehlásiť, že mnohé testy boli vylepšené a viaceré nové testy boli pridané, čím sa vyriešili problémy, ktoré sa dodatočne objavili. U vylepšených testov sme sa snažili o to, aby sa zbytočne nemazali súbory, ktoré sa mazať nemusia a aby testom neprešli súbory, ktoré bolo potrebné zmazať. Aj keď asi ťažko môžeme hovoriť o 100% úspešnosti jednotlivých testov, k vylepšeniu u jednotlivých testov určite došlo. Dôkazom toho sú napríklad testy na podobné veľkosti súborov a na správne pomenovanie súborov, ktoré ušetria stovky súborov, ktoré by sa predtým boli zmazali.

Veľmi dôležité bolo, že sme viaceré testy do programu zaradili ešte pred výber najlepšieho páru a tak sme zabránili tomu, aby sa zbytočne zahadzovali celé páry keď bola pravdepodobnosť, že do najlepšieho páru budeme môcť vybrať vhodnejší súbor. Všetky nevhodné súbory sme sa snažili odstrániť čo najskôr. A naozaj zmazaných párov bolo menej. Kým v pôvodnej verzii pred úpravami v rámci bakalárskej práce bolo zmazaných približne 950 párov, v aktuálnej verzii sa ich po výbere najlepšieho páru zmaže asi 350. Pritom si treba uvedomiť, že najlepšie páry sa vyberajú novým testom, ktorý do najlepších párov vybral až 3 098 súborov k seriálom oproti pôvodným 320. Súborov k filmom sa zachovalo 8447 oproti pôvodným 5384, čo je tiež vylepšenie. Pritom ide o počty súborov v pároch, ktoré sa zachovajú aj po záverečnom prečistení adresárovej štruktúry.

Literatúra

[1] Bojar, O; Žabokrtský, Z.: CzEng: *Czech-English Parallel Corpus*, Release version 0.5 PBML 86 (Prague Bulletin of Mathematical Linguistics), 2006.

[2] Lemay L.: *Naučte se PERL za 21 dní*, Computer Press, Praha, 2002.

[3] Logy vygenerované jednotlivými testami

[4] Specifikace XML, např. http://www.w3.org/XML

[5] Varga, D. et al.: Parallel Corpora for Medium Density Languages. In N. Nicolov, K. Bontcheva, G. Angelova and R. Mitkov (eds): Recent Advances in Natural Language Processing IV. Selected papers from RANLP-05 John Benjamins.